Heap cheatsheet for coding interviews

Introduction

A heap is a specialized tree-based data structure which is a complete tree that satisfies the heap property.

- Max heap - In a max heap, the value of a node must be greatest among the node values in its entire subtree. The same property must be recursively true for all nodes in the tree.
- Min heap - In a min heap, the value of a node must be smallest among the node values in its entire subtree. The same property must be recursively true for all nodes in the tree.

In the context of algorithm interviews, heaps and priority queues can be treated as the same data structure. A heap is a useful data structure when it is necessary to repeatedly remove the object with the highest (or lowest) priority, or when insertions need to be interspersed with removals of the root node.

Learning resources

- [Learning to Love Heaps](#), basecs
- [Heapify All The Things With Heap Sort](#), basecs
- [Heaps](#), James Aspnes, Yale University

Implementations

Language	API
C++	`std::priority_queue`
Java	`java.util.PriorityQueue`
Python	`heapq`

JavaScript	N/A

Time complexity

Operation	Big-O
Find max/min	O(1)
Insert	O(log(n))
Remove	O(log(n))
Heapify (create a heap out of given array of elements)	O(n)

Techniques

Mention of k

If you see a top or lowest *k* being mentioned in the question, it is usually a signal that a heap can be used to solve the problem, such as in [Top K Frequent Elements](#).

If you require the top *k* elements use a Min Heap of size *k*. Iterate through each element, pushing it into the heap (for python `heapq`, invert the value before pushing to find the max). Whenever the heap size exceeds *k*, remove the minimum element, that will guarantee that you have the *k* largest elements.

Essential questions

These are essential questions to practice if you're studying for this topic.
- [Merge K Sorted Lists](#)
- [K Closest Points to Origin](#)

Recommended practice questions

These are recommended questions to practice after you have studied for the topic and have practiced the essential questions.

- [Top K Frequent Elements](#)
- [Find Median from Data Stream](#)